Homework Assignment # 4

Due: Friday, April 26, 2019, 11:59 p.m.
Total Points: 100

This assignment is designed to help you better understand concepts that were presented during class. You may work in small groups (2 or 3 individuals) on the homework assignments, but it is expected that you submit your own work. Therefore, **each student is responsible for submitting their own homework**, whether they work in groups or individually. The submitted homework must include the name of other individuals you worked with (if applicable), as well as all resources that were used to solve the problem (e.g. web sites, books, research papers, other people, etc.). Academic honesty is taken seriously; for detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

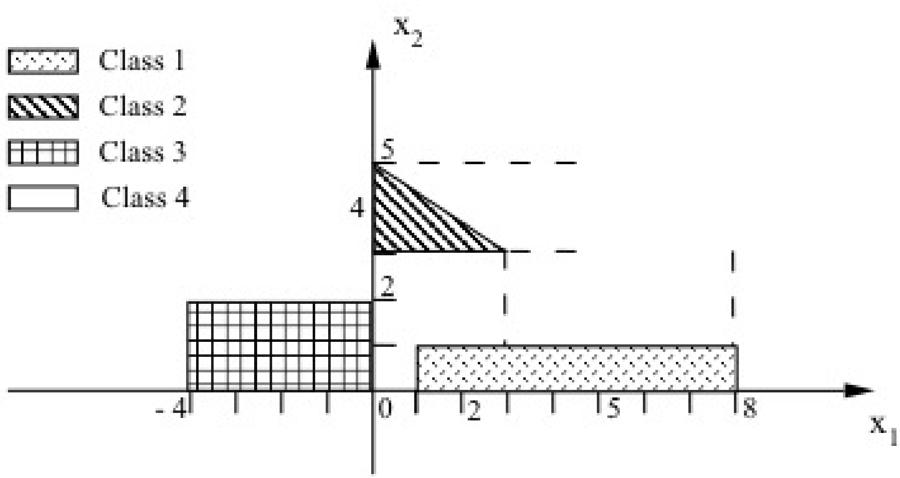
Your assignment must be submitted as a single pdf document on Canvas. The questions and your answers must be typed; for example, in Latex, Microsoft Word, Lyx, etc. Images may be scanned and inserted into the document if it is too complicated to draw them properly. Be sure to show all work.

All code (if applicable) should be turned in when you submit your assignment. Use Matlab, Python, R, Java or C. Be sure to include an complete explanation on how to run/execute your code.

All assignments must be submitted on time to receive credit. No late work will be accepted, unless you have a prior arrangement with the instructor.

Question 1. [25 POINTS]

The following figure shows the decision regions of four classes. Design a neural network that correctly classifies points from each class, using a network of M-P neurons with three output units. For Class $i\,(1 \le i \le 3)$, classification requires that $\boldsymbol{y_i = 1}$, while $\boldsymbol{y_j = -1}$ for $j \neq i$; Class 4 is recognized when $y_i = -1$ for $1 \le i \le 3$.



Question 2. [25 points]

In class, we discussed minimizing the following cost function for the K-means algorithm:

$$J = \sum_{j=1}^{K} \sum_{i \in C_j} ||\boldsymbol{x}_i - \boldsymbol{u}_j||^2 \tag{1}$$

where K is the number of clusters, i indexes the data points in cluster C_j, $\boldsymbol{x}_i$ represents the i^{th} data point, and $\boldsymbol{u}_j$ represents the mean (center) of cluster j. Suppose now, that we modify the cost function to the following:

$$J(\boldsymbol{u}_j) = \sum_{j=1}^{K} \sum_{i=1}^{N} w_{ij} ||\boldsymbol{x}_i - \boldsymbol{u}_j||^2 \tag{2}$$

where N is the number of data points, and w_{ij} is a weighting factor that is defined as follows:

$$w_{ij} = \begin{cases} 1, & \text{if they data point } \boldsymbol{x}_i \text{ lies in cluster } j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

(a) Show that the minimizing solution of this cost function is:

$$\hat{\boldsymbol{u}}_j = \frac{\sum_{i=1}^{N} w_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{N} w_{ij}} \tag{4}$$

(b) How do you interpret the expressions in the numerator and denominator of this formula?

Question 3. [25 points]

The $(\boldsymbol{x}, y)$ relationship of a Gaussian-based RBF network is defined by:

$$y(i) = \sum_{j=1}^{K} w_j(n) e^{-\frac{1}{2\sigma^2(n)}||\boldsymbol{x}_i - \boldsymbol{u}_j(n)||^2}, \quad i = 1, 2, \ldots, n \tag{5}$$

where $\boldsymbol{u}_j(n)$ is cluster center of the j^{th} Gaussian, the width $\sigma(n)$ is common to all K Gaussian components, and $w_j(n)$ is the linear weight assigned to the output of the j^{th} Gaussian component. These parameters are measured at time n. The cost function used to train the network is defined by:

$$E = \frac{1}{2} \sum_{i=1}^{n} (d(i) - y(i))^2 \tag{6}$$

where $d(i)$ is the desired output. The cost function is a convex function of the linear weights in the output layer, but non-convex with respect to the centers and width of the Gaussian units.

(a) Evaluate the partial derivatives of the cost function with respect to each of the network parameters $w_j(n)$, $\boldsymbol{u}_j(n)$, and $\sigma(n)$.

(b) Use the gradients obtained in part (a) to determine update formulas for all the network parameters, assuming learning rates of η_w, η_u, and η_σ for the network parameters, respectively. Assume that the underlying update rule is the same for convex and non-convex functions.

Question 4. [10 POINTS]

Consider a Support Vector Machine (SVM) and the following training data from two categories:

$$\begin{array}{llll} w_1: & (1,1)^T & (2,2)^T & (2,0)^T \\ w_2: & (0,0)^T & (1,0)^T & (0,1)^T \end{array}$$

(a) Plot these six training points, and construct by inspection the weight vector for the optimal hyperplane, and the optimal margin.

(b) What are the support vectors?

Question 5. [15 POINTS]

Using the expressed solution for the optimal weights (e.g. w_o), show that the optimal margin of separation between support vectors from the two classes, ρ, can be expressed in terms of the Lagrange multipliers as:

$$\rho = \frac{2}{\left(\sum_{i=1}^{N_s} \alpha_i\right)^{1/2}} \tag{7}$$

where N_s is the number of support vectors.